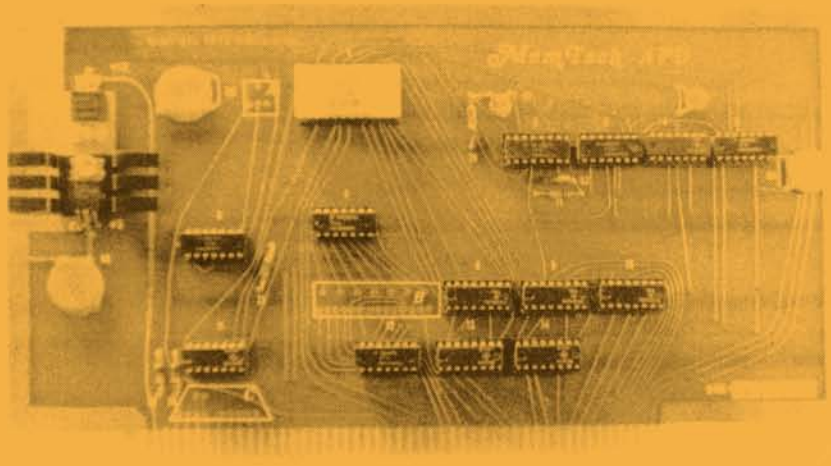
Fortran, Pascal and Basic 3 Times Faster !

Using the MemTech Fortran or Compiled Basic libraries, or Pascal MT+ with the

--MemTech ARITHMETIC PROCESSING UNIT Interface--



Interface Features:

** Floating Point multiply in 42 uS (vs 3000 uS in software)

** Full S-100 buss compatibility; on-board clock

** AMD 9511 APU chip featuring internal 16-byte stack

** Trigonometric and exponential functions (41 in all)

** Automatic floating point normalization and error codes

** Jumperable as any pair of I/O ports; interrupts or polling

** Sockets for all IC's; front and rear solder mask

Proportional speed improvements:

Statement	Pascal	Fortran	Basic
A = SIN(B)	7x faster	7x faster	7x faster
A = SQRT(B)	7x faster	25x faster	7x faster
A = B*C/D	2x faster	3x faster	2x faster

Interface prices (assembled & tested): $360 with 4MHz 9511 chip; $260 with 2MHz
MemTech Fortran library (user must already own Microsoft Fortran): Add $100
MemTech Compiled Basic library (user must own Microsoft Compiled Basic): Add $100
Both Fortran and Basic libraries: $150
MemTech Basic-M (self-contained, fully compatible with Basic-E): Add $25
New!! Pascal MT+ (trademark MT Microsystems) - ISO Pascal Compiler to Microsoft format reloc code, plus editor, pre-compiler and APU support: $470
All software is designed to operate under the CP/M operating system (trademark Digital Research); please specify 8" or 5" diskette and format

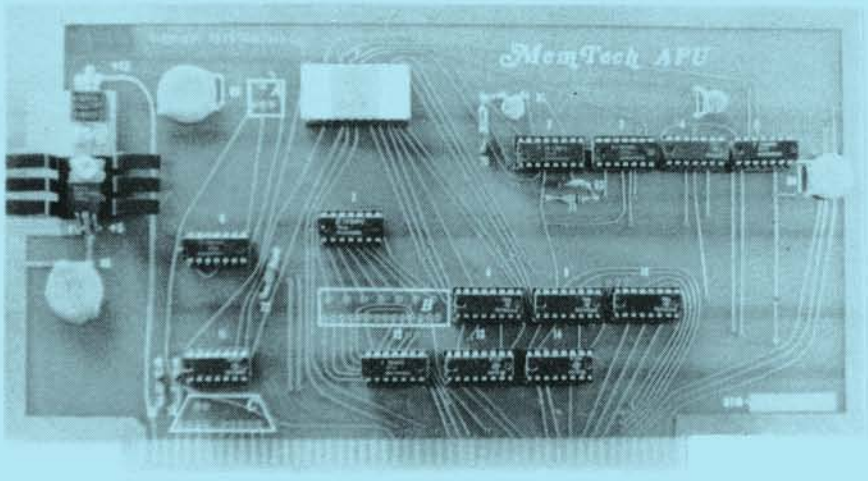
** Price includes shipping to any point in U.S.
** California residents add 6% sales tax
** Delivery time: typical one week, maximum 3 weeks

MemTech Co., P.O. Box 1548, San Bruno, CA. 94066 Tel. (415) 878-1300

Basic 3 Times Faster !

Announcing the Basic-M modification to Basic-E utilizing the Ultra-Fast

--MemTech ARITHMETIC PROCESSING UNIT Interface--



Interface Features-

** Floating Point multiply in 56 us
(vs. 3000 us in software)

** Full S-100 buss compatibility with any clock speed

** AMD 9511 APU chip featuring internal 16-byte stack

** Parallel processing of arithmetic operations while
main CPU executes the program

** Trigonometric and exponential functions (41 in all)

** Automatic floating point normalization and error codes

** Jumperable as any pair of I/O ports

** Sockets for all IC's, front and rear solder mask

Basic-M offers these execution speed improvments for Basic-E statements:

A = SIN B	---	7X faster
A = B↑C	---	7X faster
PRINT A;B	---	2X faster (with fast I/O)

Prices:

Kit without APU chip: $155 *** Add $15 for assembly ***
Add $25 for Basic-M on CP/M floppy diskette
Kit with 2 MHz APU chip: $375 --- multiply in 84us
Kit with 3 MHz APU chip: $445 --- multiply in 56us, Basic slightly faster

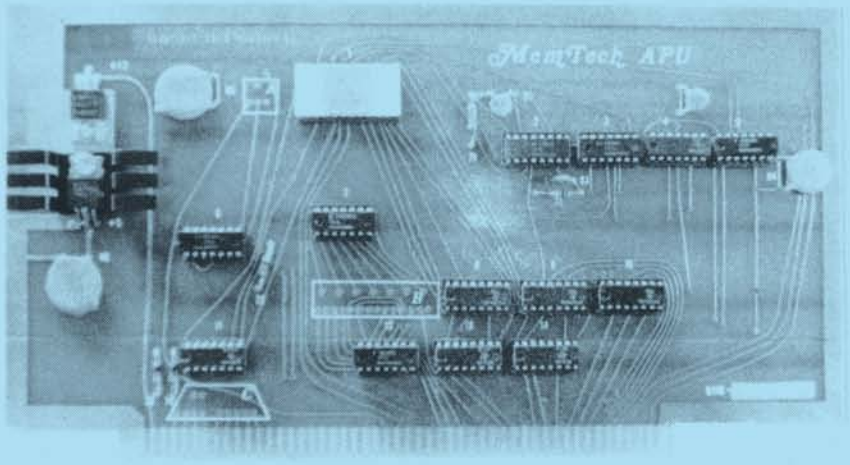
** Price includes shipping to any point on U.S.
** California residents add 6% sales tax
** Delivery time: typical one week, maximum 3 weeks

MemTech Co., 4891 Clairemont Mesa Blvd., Dept. C, San Diego, CA. 92117
Tel (714)-292-1219

Fortran and Basic 3 Times Faster !

Announcing the MemTech Fortran library and MemTech version of Basic-E
utilizing the ultra-fast

--MemTech ARITHMETIC PROCESSING UNIT Interface--



44H

Interface Features:

** Floating Point multiply performed in 56 us
(vs. 3000 us in software)

** Full S-100 buss compatibility; on-board clock

** AMD 9511 APU chip featuring internal 16-byte stack

** Trigonometric and exponential functions (41 in all)

** Automatic floating point normalization and error codes

** Jumperable as any pair of I/O ports;
interrupt operation or polling

** Sockets for all IC's; front and rear solder mask

Proportional speed improvements:

Statement	Basic	Fortran
A = SIN(B)	7x faster	7x faster
A = SQRT(B)	7x faster	25x faster
A = B*C/D	2x faster	3x faster

Prices (fully assembled and tested):
With 2 Mhz 9511 chip: $390 --- multiply in 84us Without 9511 chip: $170
With 3 Mhz 9511 chip: $460 --- multiply in 56us
Add $100 for MemTech Fortran library (user must already own Microsoft Fortran)
Add $25 for MemTech Basic (self-contained, fully compatible with Basic-E)
All software is designed to operate under the CP/M operating system
(trademark Digital Research Co.); specify 8" or 5" diskette (10/16 hole)

** Price includes shipping to any point on U.S.
** California residents add 6% sales tax
** Delivery time: typical one week, maximum 3 weeks

MemTech Co., 4891 Clairemont Mesa Blvd., San Diego, CA. 92117 / (714)292-1219

MT MemTech Co.

P.O. Box 1548
San Bruno, CA 94066
Tel. (415) 878-1300

INVENTORY REDUCTION SALE

Memtech is overstocked on AMD 9511 chips!

From now until June 30 we are offering a 15% discount on assembled APU interfaces with AMD 9511 chips.

Also, we are offering individual AMD 9511 chips at heavy discounts -- please call for quotations.

There are several new developments that our customers may wish to be aware of:

-- Our APU interface is capable of supporting the AMD 9512 chip, which offers 32/64 bit operations. There is currently no software available for it, though.

-- Our Fortran and Basic libraries now handle integer exponents up to 20 by successive multiplication.

-- Our Fortran MOD function for negative numbers now operates correctly.

-- We can provide instructions for fitting a user underflow/overflow routine into our libraries.

Customers wishing updates of their Fortran or Compiled Basic libraries should send a check for $20 to cover shipping and materials, and specify the media size/density/sectoring, as well as the command & data port numbers.

MEMTECH APU FORTRAN SUBROUTINE LIBRARY MANUAL
12-21-79 Version 1.1

Table of contents:

1. OVERVIEW, STRUCTURE, and INSTALLATION:

The Memtech Fortran Library is a supplement (not a replacement) for Microsoft Fortran or Cromemco Fortran. The user must already have purchased one of these from either Microsoft Co., 819 Two Park Central Tower, Albequerque, NM. 87108 or Cromemco Co., 280 Bernardo Ave., Mountain View, CA. 94043 or from an authorized distributor for one of the above. Use of the Memtech library with the APU interface board will result in an approximate overall 3 times speed increase for computational programs. A library to use with Microsoft Compiled Basic is available as a separate software product.

The library consists of two parts: a fortran relocatable module APUFLIB.REL containing the replaced library functions, and a program MODFLIB.COM to remove those functions from the standard Fortran library that you already own. The program MODFLIB.COM must first be copied onto a disk containing your current Fortran library (usually called FORLIB.REL) using PIP. Then run MODFLIB; it will take about 5 minutes. MODFLIB will create a new file MICFLIB.REL without damaging the old FORLIB.REL. If the words FILE ACCESS ERROR appear, either the disk is full, FORLIB.REL could not be found, or a disk I/O error occurred. If any other error messages occur, contact Memtech Co. Correct completion is signaled by the message MODIFICATION COMPLETE. After successfully running MODFLIB, any program may be loaded to use the Memtech Library by the sequence:

```
L80 PROG, APUFLIB/S, MICFLIB/S
```

If you wish to shorten this sequence, you may use CP/M and the Microsoft program LIB in the following manner:

```
REN SAVFLIB.REL=FORLIB.REL  (if you wish to save it)
LIB APUFLIB, MICFLIB /E
```

Thus the Memtech library completely replaces your old library (which is saved in SAVFLIB just in case), and program loading sequences are the same as for standard Microsoft Fortran.

The supplied APUFLIB expects the Memtech APU interface to be jumpered for the standard ports, ~~40H and 41H~~ 44 and 45 H. If your requirement is different, we will be glad to supply a special version at no extra cost.

As previously noted, the total speed improvement should be approx. 3 times faster. Benchmark tests for particular single Fortran statements are shown below. Tests were made using a 4MHz Z-80 processor and a 3MHz APU interface.

Benchmark	Microsoft library	Memtech library
DO 5 I=1,10000 5 X=SIN(1)	103 sec.	14 sec. 7.36 times faster
DO 5 I=1,10000 5 X=SQRT(3)	125 sec.	5 sec. 25.00 times faster
DO 5 I=1,10000 5 X=.3*.7	13 sec.	4 sec. 3.25 times faster
K=255 DO 5 I=1,10000 5 J=K*K	3.5 sec.	1.12 sec. 3.13 times faster

3. SUBROUTINES REPLACED:

The following single precision and integer arithmetic library modules are replaced. Double precision arithmetic functions are not performed by the APU interface.

$AA	Addition: real + integer
$AB	Addition: real + real
$D9	Division: integer / integer
$DA	Division: real / integer
$DB	Division: real / real
$M9	Multiplication: integer * integer
$MA	Multiplication: real * integer
$MB	Multiplication: real * real
$SA	Subtraction: real - integer
$SB	Subtraction: real - real
$EA	Exponentiation: real ** integer
$EB	Exponentiation: real ** real
$CA	Conversion: integer to real
$CH	Conversion: real to integer

ALOG	Natural logarithm
ALOG10	logarithm base ten
SIN	Sine
COS	Cosine
ATAN	Arc tangent
EXP	Exponential
SQRT	Square root

The following library modules are new additions:

ASIN	Arc Sine
ACOS	Arc Cosine
TAN	Tangent

The total size of your program may or may not be smaller than if it were loaded with the standard Fortran library. Chances are very good that it will be much smaller (usually by over 3000 bytes). We guarantee that it will not increase the size of your program by more than 800 bytes. The reason that the program size could turn out larger is because some Microsoft library routines call others, a relationship which we must preserve.

4. ENVIRONMENTS THAT APULIB WILL OPERATE IN:

We will distribute the Memtech Fortran Library on 4 basic floppy disk formats: A) 8" single density, standard CP/M format, B) 8" single density, Cromemco CDOS format, C) 5" Micropolis quad density, Micropolis CP/M format (Trademark Micropolis Co., 7959 Deering Ave., Canoga Park, CA. 91304), and D) 5" North Star quad density, North Star CP/M format (Trademark North Star Computers, 1440 Fourth Street, Berkely, CA. 94710).

5. NUMBER RANGE, UNDERFLOW and OVERFLOW:

The AMD 9511 Arithmetic Processor chip utilized by the APU interface represents numbers in a slightly different internal format than the Microsoft library. Basically, the 9511 offers one less bit of exponent, and one more bit of mantissa. Time taken to convert number formats is minimal since this operation is combined with that of pushing or popping operands onto the 9511 internal stack. However, in some some scientific applications there is a small possibility that a program which runs fine under the Microsoft library may experience overflow or underflow errors when run using the APU interface. If the magnitude of the result of an arithmetic operation is > 9.2 E +18 or < 2.7 E -20, APULIB will set the result to that upper/lower bound. Additionally, for overflow, the message **OV** will be sent to the operator's console (just as in the Microsoft version). Program constants which are outside the above range but within the standard Microsoft range (E+-38) will be converted to the largest or smallest 9511-range number at run-time.

6. WARRANTY:

Software is warranted against forseeable defects for 90 days after purchase. Future improvements/revisions of Memtech software will be offered to original purchasers at media cost plus a small handling fee. Users will be notified if and when such improvements become available. If any problems are found, please contact Memtech Co.

Memtech Co., 4891 Clairemont Mesa Blvd., San Diego, CA. 92117
Telephone: (714) 292-1219

serial #0018

8. USER COPY OF NON-DISCLOSURE AGREEMENT:

The following agreement is on file at Memtech Co:

The party below agrees that 1) each Memtech software purchase is for use on a single computer only, by persons authorized to use that computer, and 2) a maximum of two copies of such software may be made for backup purposes only, and 3) all such copies will display the Memtech copyright, and 4) all copies will be safeguarded against disclosure to or use by persons not authorized by Memtech to use such software.

_____ Signature _____ Date

_____ Print Name

_____ Print Company Name

_____ Print Title

_____ Print Address

_____

_____ Print Telephone Number

Memtech Co., 4891 Clairemont Mesa Blvd., San Diego, CA. 92117
Telephone: (714) 292-1219

BASIC-M

Basic-M is a modification to the Basic-E distributed with the CP/M users group library. CP/M is a trademark of Digital Research Co. The run-time module, "RUN", and the Basic compiler, "BASIC", are supplied. Only the run-time module, RUN, required changes.

Basic-E has the major advantage of compression of code in the executable form so that a much larger program can be run in the same memory space compared with by other Basic interpreters. The Basic-M modification makes RUN execute much faster for computational portions of programs. The user is held responsible for the tasks of purchasing the CP/M Operating System from Digital Research Co., Box 579, Pacific Grove, CA. 93950. Basic-E was developed by Gordon E. Eubanks, Jr. as a master's thesis in 1976. The revision of Basic-E used in Basic-M is 2.2.

The speed improvement of Basic-M is most dramatic in the execution of transcendental functions. Due to the high overhead involved in looping or conditional operations, these functions will yield the worst speed improvement comparitively. Operations which involve string manipulation will be only slightly faster and print statements which only involve character strings will be no faster. However, print statements involving variables are significantly speeded due to the decreased time necessary to convert from internal binary format to decimal using the APU interface.

Currently Basic-M is an overlaid modification of Basic-E. In other words, the modification does not reduce the size of the Basic-E module RUN. Potentially, the size could be reduced from 6K to 4K by recompiling the source of Basic-E. Memtech plans to accomplish this task at some time in the future when Basic-M has been thoroughly tested by its users.

In Basic-M, the entire floating-point package (including input-output) portion of Basic-E has been replaced by calls to the Memtech APU interface. The software assumes that the interface is jumpered for I/O port 40H & 41H. If your system already uses port 40H & 41H for another I/O device, we suggest that you first attempt to change it, so that you will be compatible with future Memtech software products. If that is not possible, we will gladly exchange your diskette for one configured for any other specified I/O ports.

Basic-M calls the Memtech APU interface (with any speed 9511) to perform all arithmetic operations at the time they are required. However, the result of such operations is not popped off the APU's internal stack until needed. Therefore, it is only necessary to loop waiting for APU operation completion just before the answer is needed, or just before another operation is initiated. In this manner, parallel processing is performed to a certain extent. Because the APU interface performs arithmetic so quickly, the microprocessor itself is the speed limiting factor while decoding the operation and moving data from place to place. Therefore, using either a 4 MHz or 2MHz 9511 chip yields approximately the same result.

All features of Basic-E are retained except that the number format is changed to one involving one bit less range, i.e. 10**20 instead of 10**40. Basic-E manuals may be obtained at your local computer store or from Tarbell Electronics, 20620 South Leapwood Ave., Suite P, Carson, CA. 90746.

The supplied 8" floppy diskette is IBM 3740 format compatible with CP/M file structure. For users with a single disk drive, the procedure for copying RUN and BASIC to another disk is: DDT RUN.COM, <change diskettes>, G0, SAVE 48 RUN.COM, <change diskettes>, DDT BASIC.COM, <change diskettes>, G0, SAVE 48 BASIC.COM.

If you experience any software problems in running Basic-M which are not known bugs in Basic-E, please contact us. You will be notified of bugs when they are solved and tested (up to 6 months after purchase).

Memtech currently is investigating modifications to Microsoft Fortran IV and UCSD Pascal. You will be notified when these programs are ready for distribution.

MemTech APU

4891 Clairemont Mesa Blvd.
San Diego, CA. 92117
Tel. (714) 292-1219

F E A T U R E S U M M A R Y

ULTRA-FAST S-100 Arithmetic Processing Board

Will perform a floating point multiply in 84uS (2-MHz 9511) or 56uS (3-MHz 9511) vs 3000uS in software.

Single Precision, Double Precision, and Floating Point

Does add, subtract, multiply, and divide in the above formats. Plus it has trig. functions also, along with log, and natural log, and many many more. 41 functions in all.

Automatic Normalization

Does automatic floating point normalization which is very costly to do in software.

2-MHz and 4-MHz

This board will run with the various CPU speeds available.

Error Codes

The APU board generates error codes such as division by zero, square root or log of a negative number, and others. It also determines whether the result is zero, and whether the operation resulted in a carry or borrow.

SOFTWARE AND HARDWARE INTERUPTS

One can jumper to any of the eight vectored interupts or the non-maskable interupt. There are two ways in which the board may be jumpered. The first will generate an interupt after the completion of any operation. The second is a software interupt control which will only generate an interupt if bit 80H is set in the command byte.

INTERNAL STACK

The board has its own internal stack of 16 bytes which it uses for its own operations and which can be used for temporary storage of variables so that the immediate results do not have to be read off. (Note this is another time saving feature).

Other Notes:

All accesses to this board are done to a pair of I/O ports; these I/O ports are jumperable to any of the 128 pairs of ports.

APU PARTS LIST

Rev. 0918

Integrated Circuits

IC1	AMD9511	High-speed arithmetic processing unit
IC2	74123	Dual retriggerable multivibrator
IC3	7400	Quad 2-input NAND gate
IC4	7432	Quad 2-input OR gate
IC5	7404	Hex inverter
IC6	7474	Dual D flip-flop
IC7	7430	8-input NAND gate
IC8	74367	Hex tri-state bus driver
IC9	"	"
IC10	"	"
IC11	74123	Dual retriggerable multivibrator
IC12	7405	Hex open collector inverter
IC13	7404	Hex inverter
IC14	7475	Quad latch
IC15	"	"

Capacitors (all ceramic disc 20v or better)

C80 - .1uF	C86 - 18pF matched
C81 - 100pF	C87 - 50pF
C82 - 1500pF	C88 - 100pF
C83 - .001uF	
C84 - .002uF	
C85 - .1uF	

Resistors (all 1/4 watt carbon 5% or better)

R70 - 22k ohm (red-red-orange)
R71 - 10k ohm (brown-black-orange)
R72 - 3k ohm (orange-black-red)
R73 - matched (15k=2MHz or 8.2k=3MHz) OR 4.7k with trim pot
R74 - 25k ohm trim pot (optional)
R75 - 3k ohm (orange-black-red)
R76 - 4.7k ohm (yellow-violet-red)
R77 - 3k ohm (orange-black-red)
R78 - 3k ohm (orange-black-red)

Misc.

LM340T-5 (7805) +5 volt regulator
LM340T-12 (7812) +12 volt regulator
Printed circuit board
Sockets for all IC's
Heat sink
2 sets of 8-32 by 1/4" screws & nuts

APU ASSEMBLY INSTRUCTIONS

SOLDERING:

Use a good resin-core solder (acid-core will corrode). Use a small soldering element, preferably about 25 watts. Keep your tip clean by wiping it on a sponge. Apply heat first to the joint, then apply solder, then remove the solder, then remove the heat. Don't apply heat to any one joint for more than a few seconds at a time, since the clad will eventually seperate from the board under too much heat. Be sure that there is a smooth flow of solder over the entire connection (front & back), and that the joint looks shiny.

PARTS:

Check the parts enclosed against the parts list to make sure there is nothing missing. Insert each IC socket into an appropriate 14,16 or 24 pin location. All parts are mounted on the lettering side of the board. An IC socket is normally placed so that pin 1 of the IC will plug into the corner with the notch. Pin 1 on this board is always in the lower left corner when viewing the lettering "MemTech APU" face up and upright. Solder each lead of the socket after it is in place from the rear of the board. DO NOT install sockets in the 2 extra positions with double holes.

Bend the leads of the resistors to fit in each marked location. Locations are marked with a solid line between two holes and a number from 70-73. R74 is the 3-lead trim pot (see jumpering instructions before you install). When the resistor is inserted so that it is flush against the board, cut the leads on the back side so that they are almost flush and then solder (careful - the resistor may get hot).

Capacitors should be mounted similarly. If they are so large as to be in the way, leave a little lead on the front side of the board and bend them carefully so that the lie flat. Capacitor locations are 80-88.

Mount the voltage regulators in their proper places alongside the +5 and +12 lettering. (consult the parts list for identification). The heat sink should be placed under the +5 volt regulator and the screws should be inserted from the rear of the board. Be sure to solder all 3 leads.

APU JUMPERING INSTRUCTIONS

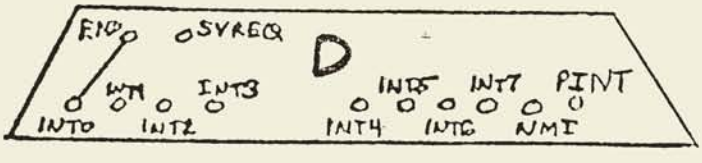
CLOCK:

The 9511 chip requires a clock signal between 2MHz and 3MHz, depending on the part number shipped with your order. AM9511DC-1 is 3MHz, AM9511DC is 2MHz. This clock speed is a minimum and it doesn't mean that the 9511 will not run at a faster speed. There are numerous ways to jumper this board, but in essence there are two main ways. The clock is changed by changing jumper area A, jumper area B, or R73 and R74. For the two main options areas A and B need not be altered. Option 1 is to have a standard 2 or 3MHz clock for the 9511. This is accomplished by placing a jumper between the left and center pads of R74 and putting the matching 15K or 8.2K resistor in R73. As previously mentioned the 2 or 3MHz clock rate is a guaranteed minimum and the 9511 will probably operate faster than that particular speed. Should you be willing to experiment with your particular 9511 you may place the 4.7K resistor in R73 and the 25K trim-pot in R74. You should start with a resistance value close to the matching resistor supplied and slowly decrease that value while checking the more time consuming functions such as PWR or ACOS for failure.

The other options are to first alter area B by cutting the connecting pads in the back and jumpering the left pad to the center pad. This will give the 9511 a signal of PHI-2 / 2 which will be a 1MHz or 2MHz signal depending on whether your CPU is a 2MHz or 4MHz CPU. Other options are to cut the clad between the center and top pads of area A and to jumper the top pad to the left pad for PHI-2 or to the right pad for 2-MHz from pin 49 of the CPU.

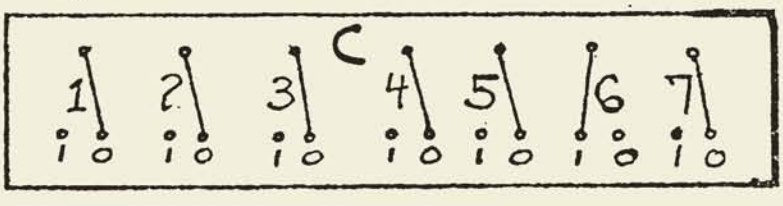
INTERRUPT:

Either the fixed END or the software-selectable SVREQ lines coming from the 9511 may be jumpered to create an interrupt. Below is the jumper configuration for the end of any command operation to cause vectored interrupt 0.



ADDRESS:

The I/O port address is jumpered in area A. Bit 1 is on the left; bit 7 on the right. Bit 0 determines whether one adresses the command port or the data port, and is therefore not jumperable. A jumper wire indicates a bit value of 0 if it leads to the right or 1 if it leads to the left. Below is the jumper configuration for I/O ports 40H and 41H (64 and 65 decimal). Your board comes pre-jumpered as shown below, and the rear clad must be cut for any change.



INSERTION OF IC's EXCEPT FOR THE APU:

If necessary, carefully bend each lead slightly inward (keeping the pin straight) in order to fit well. Insert the IC's into their appropriate sockets numbered 2-15. Do not bend the leads too many times as they can break.

HANDLING OF THE APU:

The 9511 chip is an N-channel MOS device. While recently devleoped IC's have some overvoltage protection, MOS is still damageable by static charges.
Since the voltage is what matters and not the current, handling is important until the chip is fully inserted. Do not move around on a rug or rub material against anything while holding the APU. Touch the leads as little as possible. Insert the 9511 into its 24-pin socket, being very careful that the pins do not fold underneath. Check all other IC's for folded pins.

CLEANING:

After you finish soldering, there will be many small conductive particles on the board which you cannot always see. Take a small pointed instrument (such as a screwdriver) and scrape between any printed wiring which is very close together. Visually inspect all connections for overflow of solder to a nearby one. Solder resin may be removed is desired with rubbing alcohol and a Q-tip.

MAXIMUM RATINGS above which useful life may be impaired

Storage Temperature	–65°C to +150°C
Ambient Temperature Under Bias	–55°C to +125°C
VDD with Respect to VSS	–0.5V to +15.0V
VCC with Respect to VSS	–0.5V to +7.0V
All Signal Voltages with Respect to VSS	–0.5V to +7.0V
Power Dissipation	2.0W

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

OPERATING RANGE

Part Number	Ambient Temperature	VSS	VCC	VDD
Am9511DC Am9511-4DC	$0°C \leq T_A \leq +70°C$	0V	+5.0V ± 5%	+12V ± 5%
Am9511DM	$-55°C \leq T_A \leq +125°C$	0V	+5.0V ± 10%	+12V ± 10%

ELECTRICAL CHARACTERISTICS Over Operating Range (Note 1)

Parameters	Description	Test Conditions	Min.	Typ.	Max.	Units
VOH	Output HIGH Voltage	IOH = –200μA	3.7			Volts
VOL	Output LOW Voltage	IOL = 3.2mA			0.4	Volts
VIH	Input HIGH Voltage		2.0		VCC	Volts
VIL	Input LOW Voltage		–0.5		0.8	Volts
IIX	Input Load Current	$VSS \leq VI \leq VCC$			±10	μA
IOZ	Data Bus Leakage	VO = 0.4V			–100	μA
		VO = VCC			100	
ICC	VCC Supply Current	$T_A = +25°C$		55		mA
		$T_A = 0°C$				
		$T_A = -55°C$				
IDD	VDD Supply Current	$T_A = +25°C$		55		mA
		$T_A = 0°C$				
		$T_A = -55°C$				
CO	Output Capacitance	fc = 1.0MHz, Inputs = 0V		8	10	pF
CI	Input Capacitance			5	8	pF
CIO	I/O Capacitance			10	12	pF

~~INTERFACE~~ SIGNAL DESCRIPTION

VCC: +5 Volt power supply

VDD: +12 Volt power supply

VSS: Ground

CLK (Clock, Input)

An external, TTL compatible, timing source is applied to the CLK pin.

RESET (Reset, Input)

The active high reset signal provides initialization for the chip. RESET also terminates any operation in progress. RESET clears the status register and places the Am9511 into the idle state. Stack contents and command registers are not affected.

$\overline{CS}$ (Chip Select, Input)

$\overline{CS}$ is an active low input signal which selects the Am9511 and enables communication with the data bus.

C/$\overline{D}$ (Command/Data, Input)

In conjunction with the $\overline{RD}$ and $\overline{WR}$ signals, the C/$\overline{D}$ control line establishes the type of communication that is to be performed with the Am9511 as shown below:

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	Function
0	1	0	Enter data byte into stack
0	0	1	Read data byte from stack
1	1	0	Enter command
1	0	1	Read status

$\overline{RD}$ (Read, Input)

This active low input indicates that data or status is to be read from the Am9511 if $\overline{CS}$ is low.

$\overline{WR}$ (Write, Input)

This active low input indicates that data or a command is to be written into the Am9511 if $\overline{CS}$ is low.

$\overline{EACK}$ (End Acknowledge, Input)

This active low input clears the end of execution output signal ($\overline{END}$). If $\overline{EACK}$ is tied low, the $\overline{END}$ output will be a pulse that is one clock wide.

$\overline{SVACK}$ (Service Acknowledge, Input)

This active low input clears the service request output (SVREQ).

$\overline{END}$ (End Execution, Output)

This active low, open-drain output indicates that execution of the previously entered command is complete. It can be used as an interrupt request and is cleared by $\overline{EACK}$, RESET or any read or write access to the Am9511.

SVREQ (Service Request, Output)

This active high output signal indicates that command execution is complete and that post execution service was requested in the previous command byte. It is cleared by $\overline{SVACK}$, the next command output to the device, or by RESET.

$\overline{PAUSE}$ (Pause, Output)

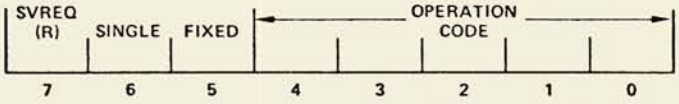
This active low output indicates that the Am9511 is unable to accept communication with the data bus. When an attempt is made to read data, write data or to enter a new command while the Am9511 is executing a command, $\overline{PAUSE}$ goes low until execution of the current command is complete. (See Pause Operation, p. 5).

DB0-DB7 (Bidirectional Data Bus, I/O)

These eight bidirectional lines provide for transfer of commands, status and data between the Am9511 and the CPU. The Am9511 can drive the data bus only when $\overline{CS}$ and $\overline{RD}$ are low.

COMMAND STRUCTURE

Each command entered into the Am9511 consists of a single 8-bit byte having the format illustrated below:



Bits 0-4 select the operation to be performed as shown in the table. Bits 5-6 select the data format appropriate to the selected operation. If bit 5 is a 1, a fixed point data format is specified. If bit 5 is a 0, floating point format is specified. Bit 6 selects the precision of the data to be operated upon by fixed point commands only (if bit 5 = 0, bit 6 must be 0). If bit 6 is a 1, single-precision (16-bit) operands are assumed. If bit 6 is a 0, double-precision (32-bit) operands are indicated. Results are undefined for all illegal combinations of bits in the command byte. Bit 7 indicates whether a service request is to be issued after the command is executed. If bit 7 is a 1, the service request output (SVREQ) will go high at the conclusion of the command and will remain high until reset by a low level on the service acknowledge pin ($\overline{SVACK}$) or until completion of execution of the succeeding command where service request (bit 7) is 0. Each command issued to the Am9511 requests post execution service based upon the state of bit 7 in the command byte. When bit 7 is a 0, SVREQ remains low.

COMMAND SUMMARY

Command Code								Command Mnemonic	Command Description (1)
7	6	5	4	3	2	1	0		
FIXED POINT SINGLE PRECISION									
R	1	1	0	1	1	0	0	SADD	Adds TOS to NOS. Result to NOS. Pop Stack
R	1	1	0	1	1	0	1	SSUB	Subtracts TOS from NOS. Result to NOS. Pop Stack
R	1	1	0	1	1	1	0	SMUL	Multiplies NOS by TOS. Result to NOS. Pop Stack
R	1	1	0	1	1	1	1	SDIV	Divides NOS by TOS. Result to NOS. Pop Stack
FIXED POINT DOUBLE PRECISION									
R	0	1	0	1	1	0	0	DADD	Adds TOS to NOS. Result to NOS. Pop Stack
R	0	1	0	1	1	0	1	DSUB	Subtracts TOS from NOS. Result to NOS. Pop Stack
R	0	1	0	1	1	1	0	DMUL	Multiplies NOS by TOS. Result to NOS. Pop Stack
R	0	1	0	1	1	1	1	DDIV	Divides NOS by TOS. Result to NOS. Pop Stack
FLOATING POINT									
R	0	0	1	0	0	0	0	FADD	Adds TOS to NOS. Result to NOS. Pop Stack
R	0	0	1	0	0	0	1	FSUB	Subtracts TOS from NOS. Result to NOS. Pop Stack
R	0	0	1	0	0	1	0	FMUL	Multiplies NOS by TOS. Result to NOS. Pop Stack
R	0	0	1	0	0	1	1	FDIV	Divides NOS by TOS. Result to NOS. Pop Stack
DERIVED FLOATING POINT FUNCTIONS (2)									
R	0	0	0	0	0	0	1	SQRT	Square Root of TOS. Result in TOS.
R	0	0	0	0	0	1	0	SIN	Sine of TOS. Result in TOS.
R	0	0	0	0	0	1	1	COS	Cosine of TOS. Result in TOS.
R	0	0	0	0	1	0	0	TAN	Tangent of TOS. Result in TOS
R	0	0	0	0	1	0	1	ASIN	Inverse Sine of TOS. Result in TOS.
R	0	0	0	0	1	1	0	ACOS	Inverse Cosine of TOS. Result in TOS.
R	0	0	0	0	1	1	1	ATAN	Inverse Tangent of TOS. Result in TOS.
R	0	0	0	1	0	0	0	LOG	Common Logarithm (base 10) of TOS. Result in TOS.
R	0	0	0	1	0	0	1	LN	Natural Logarithm (base e) of TOS. Result in TOS.
R	0	0	0	1	0	1	0	EXP	Exponential (e^X) of TOS. Result in TOS.
R	0	0	0	1	0	1	1	PWR	NOS raised to the power in TOS. Result to NOS. Pop Stack.
DATA MANIPULATION COMMANDS (3)									
R	0	0	0	0	0	0	0	NOP	No Operation
R	0	0	1	1	1	1	1	FIXS	Converts TOS from floating point to single precision fixed point format.
R	0	0	1	1	1	1	0	FIXD	Converts TOS from floating point to double precision fixed point format.
R	0	0	1	1	1	0	1	FLTS	Converts TOS from single precision fixed point to floating point format.
R	0	0	1	1	1	0	0	FLTD	Converts TOS from double precision fixed point to floating point format.
R	1	1	1	0	1	0	0	CHSS	Changes sign of single precision fixed point operand on TOS.
R	0	1	1	0	1	0	0	CHSD	Changes sign of double precision fixed point operand on TOS.
R	0	0	1	0	1	0	1	CHSF	Changes sign of floating point operand on TOS.
R	1	1	1	0	1	1	1	PTOS	Push single precision fixed point operand on TOS to NOS.
R	0	1	1	0	1	1	1	PTOD	Push double precision fixed point operand on TOS to NOS.
R	0	0	1	0	1	1	1	PTOF	Push floating point operand on TOS to NOS.
R	1	1	1	1	0	0	0	POPS	Pop single precision fixed point operand from TOS. NOS becomes TOS.
R	0	1	1	1	0	0	0	POPD	Pop double precision fixed point operand from TOS. NOS becomes TOS.
R	0	0	1	1	0	0	0	POPF	Pop floating point operand from TOS. NOS becomes TOS.
R	1	1	1	1	0	0	1	XCHS	Exchange single precision fixed point operands TOS and NOS.
R	0	1	1	1	0	0	1	XCHD	Exchange double precision fixed point operands TOS and NOS.
R	0	0	1	1	0	0	1	XCHF	Exchange floating point operands TOS and NOS.
R	0	0	1	1	0	1	0	PUPI	Push floating point constant "π" onto TOS. Previous TOS becomes NOS.

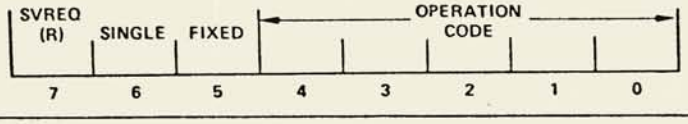
Notes: 1. NOMENCLATURE: TOS is Top Of Stack. NOS is Next On Stack.

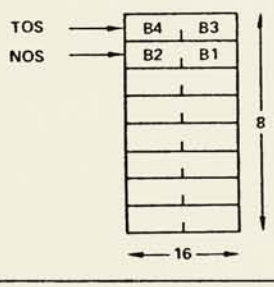
2. All derived floating point functions destroy the contents of the stack. Only the result can be counted on to be valid upon command completion.

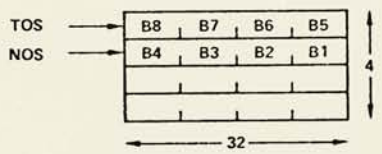
3. Format conversion commands (FIXS, FIXD, FLTS, FLTD) require that floating point data format be specified (command bits 5 and 6 must be 0).

COMMAND STRUCTURE

Each command entered into the Am9511 consists of a single 8-bit byte having the format illustrated below:







SOFTWARE CONSIDERATIONS

The Memtech APU interface is jumperable to any two I/O ports starting with an even number. The first is the command port and the second the data port. The possible options are:

1) A write to the data port pushes data onto the APU's internal stack **(in reverse order, of course).**
2) A write to the command port initiates a command. One may determine that the command is completed either by polling the status (see below) or by using an interrupt.
3) A read from the command port retrieves the status word, which includes busy information (bit 7 on means the APU is busy with a command) and error codes.
4) A read from the data port pops data off the APU's internal stack.

A smart program will utilize all 16 bytes of the APU's internal stack to store temporary calculation results.

Contrary to the statements in the 9511 data sheet (parts of which are included in this manual), it is <u>necessary</u> to poll (or wait for interrupt) the APU until it is no longer busy processing a command before one can read the answer or initiate a second command. A smart program, however, will save time by only polling the APU when it actually needs to for the next task.

WAIT STATES

A series of wait states is generated by an I/O port read. The number of wait states depends on the speed of the central processor and the length of the PAUSE signal provided by the 9511. For a 2 MHz APU and a 2 MHz 8080 CPU, approximately 4 wait states are generated. The PAUSE signal does not occur on an I/O port write, and therefore it generates no wait states.

Modifications for the TDL ZPU:

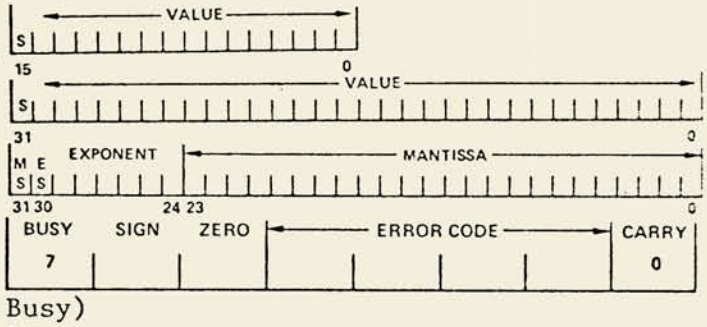
The TDL Z-80 processor card has logic which (incorrectly) prevents wait states on I/O cycles. For the APU interface to operate with this board, the clad leading from pin 5 of IC21 on the TDL ZPU board must be cut, and pin 5 should be jumpered to pin 14 (+5 volts).

MEMTECH APU PROGRAMMER'S AID

COMMAND	W/INT	NO/INT	TIME (uS) 2MHz
SADD	EC	6C	8.5
SSUB	ED	6D	15.0
SMUL	EE	6E	46.0
SDIV	EF	6F	46.0
DADD	AC	2C	10.5
DSUB	AD	2D	19.0
DMUL	AE	2E	104.0
DDIV	AF	2F	104.0
FADD	90	10	28.0-175.0
FSUB	91	11	29.0-176.0
FMUL	92	12	84.0
FDIV	93	13	85.5
SQRT	81	01	400.0
SIN	82	02	2232.0
COS	83	03	2059.0
TAN	84	04	2877.0
ASIN	85	05	3834.0
ACOS	86	06	3867.0
ATAN	87	07	3003.0
LOG	88	08	2245.0
LN	89	09	2239.0
EXP	8A	0A	2308.0
PWR	8B	0B	4646.0

COMMAND	W/INT	NO/INT	TIME (uS) 2MHz
NOP	80	00	2.0
FIXS	9F	1F	46.0-108.0
FIXD	9E	1E	50.0-173.0
FLTS	9D	1D	48.0- 93.0
FLTD	9C	1C	48.0-189.0
CHSS	F4	74	13.0
CHSD	B4	34	17.0
CHSF	95	15	8.0
PTOS	F7	77	8.0
PTOD	B7	37	10.0
PTOF	97	17	10.0
POPS	F8	78	5.0
POPD	B8	38	6.0
POPF	98	18	6.0
XCHS	F9	79	9.0
XCHD	B9	39	13.0
XCHF	99	19	13.0
PUPI	9A	1A	8.0

SINGLE PRECISION
-32,768 to +32,767
DOUBLE PRECISION
-2,147,483,648 to +2,147,483,647
FLOATING POINT
$\pm(2.7 \times 10^{-20}$ to $9.2 \times 10^{18})$ and zero



STATUS BYTE:
BUSY: Currently executing a command (1 = Busy)
SIGN: Value at TOS is negative (1 = Negative)
ZERO: Value on TOS is zero (1 = Value is zero)
ERROR CODE FIELD:
0000 - No error
1000 - Divide by zero
0100 - Square root or log of negative number
1100 - Argument of inverse sine, cosine, or e^x too large
XX10 - Underflow
XX01 - Overflow
CARRY: Previous operation resulted in carry or borrow from most significant bit. (1 = Carry/Borrow, 0 = No Carry/No Borrow)

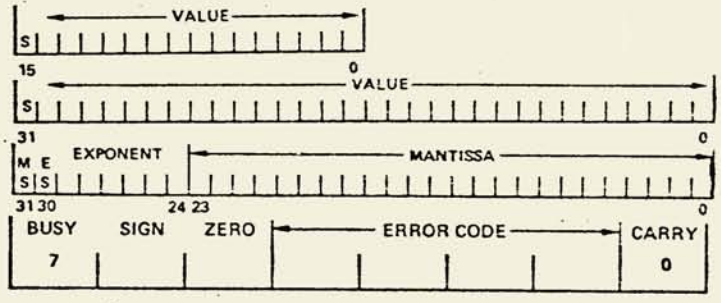
MEMTECH APU PROGRAMMER'S AID

COMMAND	W/INT	NO/INT	TIME (uS) 3 MHz
SADD	EC	6C	5.67
SSUB	ED	6D	10.00
SMUL	EE	6E	30.67
SDIV	EF	6F	30.67
DADD	AC	2C	7.00
DSUB	AD	2D	12.67
DMUL	AE	2E	69.33
DDIV	AF	2F	69.33
FADD	90	10	18.7-116.7
FSUB	91	11	32.7-117.3
FMUL	92	12	56.00
FDIV	93	13	57.00
SQRT	81	01	266.7
SIN	82	02	1488.0
COS	83	03	1372.7
TAN	84	04	1918.0
ASIN	85	05	2556.0
ACOS	86	06	2578.0
ATAN	87	07	2002.0
LOG	88	08	1496.7
LN	89	09	1492.7
EXP	8A	0A	1538.7
PWR	8B	0B	3097.3
NOP	80	00	1.3

COMMAND	W/INT	NO/INT	TIME (uS) 3 MHz
FIXS	9F	1F	30.7- 72.0
FIXD	9E	1E	33.3-115.3
FLTS	9D	1D	32.7- 62.0
FLTD	9C	1C	32.0-126.0
CHSS	F4	74	8.67
CHSD	B4	34	11.33
CHSF	95	15	5.33
PTOS	F7	77	5.33
PTOD	B7	37	6.67
PTOF	97	17	6.67
POPS	F8	78	3.33
POPD	B8	38	4.00
POPF	98	18	4.00
XCHS	F9	79	6.00
XCHD	B9	39	8.67
XCHF	99	19	8.67
PUPI	9A	1A	5.33

SINGLE PRECISION
-32,768 to +32,767
DOUBLE PRECISION
-2,147,483,648 to +2,147,483,647
FLOATING POINT

$\pm(2.7 \times 10^{-20}$ to $9.2 \times 10^{18})$ and zero



STATUS BYTE:
BUSY: Currently executing a command (1 = Busy)
SIGN: Value at TOS is negative (1 = Negative)
ZERO: Value on TOS is zero (1 = Value is zero)
ERROR CODE FIELD:
0000 - No error
1000 - Divide by zero
0100 - Square root or log of negative number
1100 - Argument of inverse sine, cosine, or e^x too large
XX10 - Underflow
XX01 - Overflow
Carry: Previous operation resulted in carry or borrow from most significant bit. (1 = Carry/Borrow, 0 = No Carry/No Borrow)

APU TEST PROGRAMS

The following two test programs are designed to verify the main functions of the interface< and only utilize the single-precision integer add function of the APU.

The first is written in BASIC-E and prints out the result of adding 2 and 3. Since numbers are popped off the APU stack in reverse order, the answer output should be 0 5 . *NOTE: This program will not work under BASIC-M since BASIC-M doesn't allow output to the APU.

The second is an equivalent program written in assembler to perform the same function. However, the answer is put into memory locations starting at 120H.

In both programs it is assumed that the APU board is jumpered for I/O locations 40H and 41H.

```
10 REM  APU TEST PROGRAM # 1
20 OUT 65,3
30 OUT 65,0
40 OUT 65,2
50 OUT 65,0
60 OUT 64,108
70 I=INP(64)
80 IF I>=128 THEN GOTO 70
90 PRINT INP(65);INP(65)
95 END
```

```
* MEMTECH APU TEST # 2
*
E000 =          MON    EQU   0E000H   ;LOCATION OF MONITOR OR CPM
0040 =          APUC   EQU   40H      ;APU COMMAND PORT
0041 =          APUD   EQU   APUC+1   ;APU DATA PORT
0100                   ORG   100H     ;START OF PROGRAM
0100 C3,30,01          JMP   START    ;JUMP OVER WORK AREA
0110            W      ORG   110H     ;START OF WORK AREA
0110 6C         COM    DB    6CH      ;COMMAND BYTE
0111 04         NBYT   DB    4        ;# OF DATA BYTES
0112 03,00,02,C0DATA   DB    3,0,2,0  ;INPUT DATA
011E =          SNBYT  EQU   W+0EH    ;TEMPORARY STORAGE
011F =          STAT   EQU   W+0FH    ;PLACE TO SAVE APU STATUS
0120 =          ANS    EQU   W+10H    ;PLACE TO STORE ANSWER
0130            START  ORG   W+20H
0130 3A,11,01          LDA   NBYT     ;GET # DATA BYTES
0133 32,1E,01          STA   SNBYT    ;SAVE
0136 21,12,01          LXI   H,DATA   ;GET LOC. FIRST DATA BYTE
0139 7E         LOOP1  MOV   A,M      ;GET DATA BYTE
013A D3,41             OUT   APUD     ;GIVE TO APU
013C 3A,1E,01          LDA   SNBYT    ;UPDATE COUNT
013F 3D                DCR   A
0140 CA,4A,01          JZ    DONE1    ;JUMP IF ALL BYTES GIVEN
0143 32,1E,01          STA   SNBYT
0146 23                INX   H        ;UPDATE POINTER
0147 C3,39,01          JMP   LOOP1
014A 3A,10,01   DONE1  LDA   COM      ;GET COMMAND
014D D3,40             OUT   APUC     ;GIVE TO APU
014F DB,40      LOOP2  IN    APUC     ;GET APU STATUS
0151 32,1F,01          STA   STAT     ;SAVE STATUS FOR VIEWING
0154 E6,80             ANI   80H      ;SEE IF APU IS READY
0156 C2,4F,01          JNZ   LOOP2    ;LOOP IF NOT READY
0159 3A,11,01          LDA   NBYT     ;GET # BYTES AGAIN
015C 32,1E,01          STA   SNBYT    ;SAVE
015F 21,20,01          LXI   H,ANS    ;GET ADDR. TO STORE ANSWER
0162 DB,41      LOOP3  IN    APUD     ;READ APU ANSWER BYTE
0164 77                MOV   M,A      ;SAVE IN MEMORY
0165 3A,1E,01          LDA   SNBYT    ;UPDATE BYTE COUNT
0168 3D                DCR   A
0169 CA,73,01          JZ    DONE2    ;JUMP IF DONE READING ANSWER
016C 32,1E,01          STA   SNBYT
016F 23                INX   H        ;UPDATE POINTER
0170 C3,62,01          JMP   LOOP3
0173 C3,00,E0   DONE2  JMP   MON      ;RETURN TO MONITOR OR CPM
0176                   END   100
```

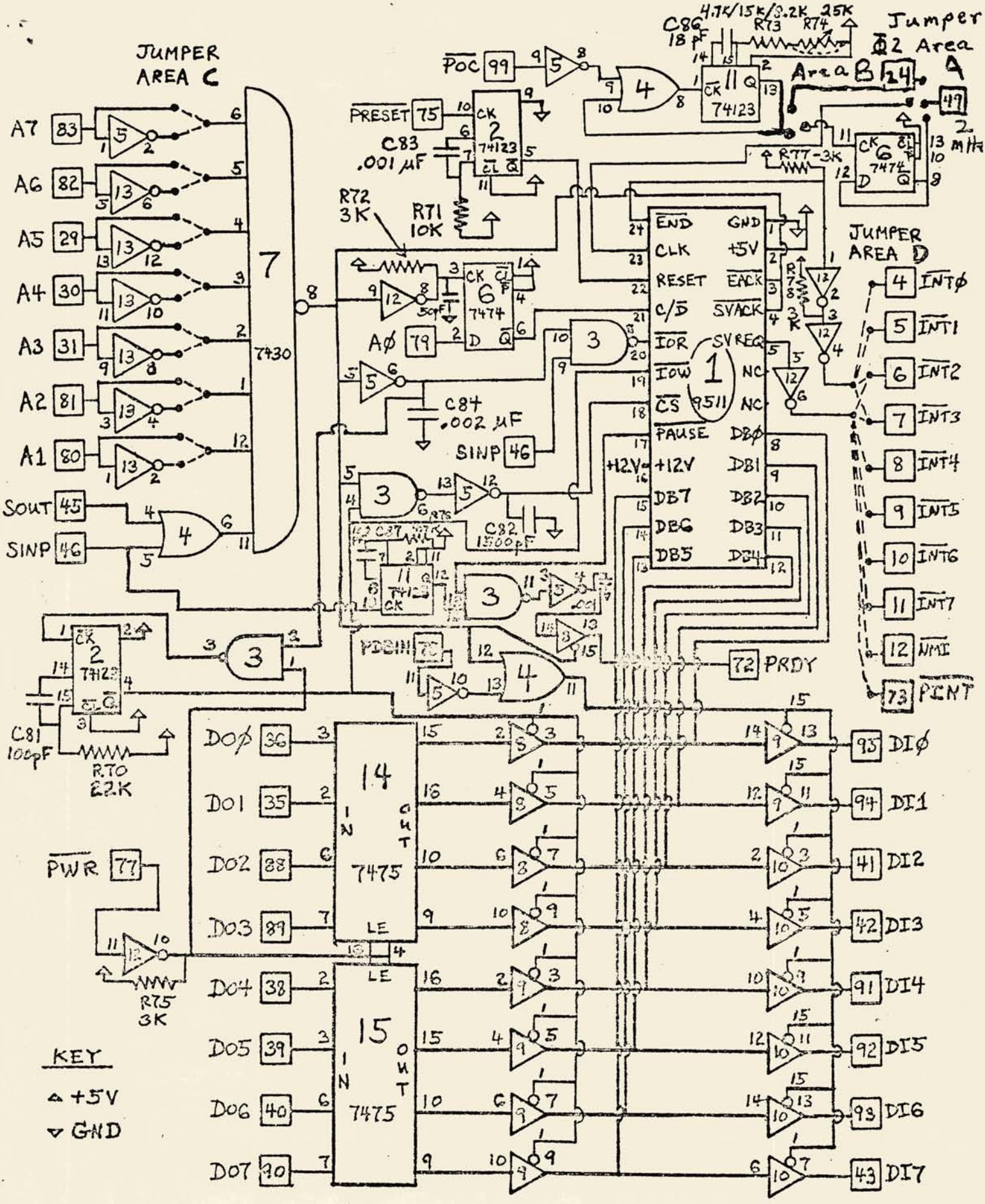
Memtech Co.

Statement of Warranty

Assembled products only are warranted to be fully operational up to 90 days after the date of purchase. Individual kit parts are also warranted for 90 days.

Repair Policy

Products purchased from Memtech in assembled or kit form may be sent postpaid to Memtech along with $25 for repair services. Memtech reserves the right to refuse repair of any product. There will be no additional repair charge unless the total retail cost of all parts replaced exceeds $5. Memtech will return the repaired product postpaid to the customer after repair is complete.



4891 Clairemont Mesa Blvd.
San Diego, CA. 92117
Tel. (714) 292-1219

MemTech APU

REV. 0518